

1

# SYSTEM AND METHOD FOR VERIFYING SEQUENCE OF INSTRUCTIONS OF SOFTWARE

## CROSS-REFERENCE TO RELATED APPLICATIONS

This application is a continuation of U.S. application Ser. No. 14/733,232, filed Jun. 8, 2015, the contents of which are incorporated by reference herein in their entirety.

## BACKGROUND

Agreements between entities, such as persons and businesses, underpin a wide variety of transactions, particularly in commercial settings. Agreements that create legally enforceable obligations are known as contracts. A smart contract is a machine that facilitates implementation of at least a portion of an act associated with an agreement between entities. For example, a vending machine can be a realization of a smart contract between a purchaser and an owner of the merchandise inside the vending machine. A smart contract can also be realized using computer hardware, software, or a combination of both. Because portions of acts associated with agreements between entities can be automated by computerized systems, advocates of smart contracts believe that the transactional costs of smart contracts may be less than the transactional costs of traditional contracts.

Another example of a smart contract is a starter interrupt device, which allows a lender at a remote location to disable an automobile whose owner is late making a payment on a loan. More generally, aspects of smart contracts can be found in computer-implemented systems and software that provide quality of service mechanisms associated with service level agreements for packet switched computer networks, automated digital rights management for copyright licenses, cryptographic systems such as those that are often used for financial transactions, and automated enforcement mechanisms in peer-to-peer file sharing networks.

## SUMMARY

Aspects disclosed herein relate generally to systems and methods for executing software, and particularly to systems and methods for executing software that includes a smart contract.

An exemplary aspect can be directed to a method for executing software. A result produced from an execution of a copy of the software by a corresponding platform can be received from each of a plurality of platforms. A number of matching results of execution of the software can be determined. Whether the number of matching results is equal to or greater than a threshold can be determined. The matching results can be accepted as an agreed-upon output of the software in response to the number of the matching results being equal to or greater than the threshold. The software can include an implementation of at least a portion of an act associated with an agreement between at least two entities.

Another exemplary aspect can be directed to a device for causing software to be executed. The system can include communication circuitry, a memory, a processor, and a bus. The communication circuitry can be configured to receive, from each of a plurality of platforms, a result produced from an execution of a copy of the software by a corresponding platform. The memory can be configured to store results from the plurality of platforms. The processor can be con-

2

figured to determine a number of matching results of execution of the software and to determine that the number of the matching results is equal to or greater than the first threshold. The bus can be coupled to the communication circuitry, the memory, and the processor.

Yet another exemplary aspect can be directed to a non-transitory computer-readable storage medium. The storage medium can include instructions that, if executed by one or more processors of a computing system, can cause the computing system to perform operations to execute software. The operations can include receiving, from each of a plurality of platforms, a result produced from an execution of a copy of the software by a corresponding platform. The operations can include determining a number of matching results of execution of the software. The operations can include determining that the number of the matching results is equal to or greater than a threshold.

Additional features, advantages, and aspects of the disclosed subject matter may be set forth or apparent from consideration of the following detailed description, drawings, and claims. Moreover, it is to be understood that both the foregoing summary and the following detailed description are illustrative and are intended to provide further explanation without limiting the scope of the claims.

## BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are included to provide a further understanding of the disclosed subject matter, are incorporated in and constitute a part of this specification. The drawings also illustrate aspects of the disclosed subject matter and together with the detailed description serve to explain the principles of aspects of the disclosed subject matter. No attempt is made to show structural details in more detail than may be necessary for a fundamental understanding of the disclosed subject matter and various ways in which it may be practiced.

FIG. 1 is a diagram illustrating an example of a system in which software can be executed.

FIG. 2 is a diagram illustrating an example of a system in which software can be executed.

FIG. 3 is a diagram illustrating an example of a system in which software can be executed.

FIG. 4 is a diagram illustrating an example of a system in which software can be executed.

FIG. 5 is a diagram illustrating an example of a system in which software can be executed.

FIG. 6 is a diagram illustrating an example of a platform illustrated in FIGS. 1 through 5.

FIG. 7 is a flow diagram illustrating an example of a method for executing software.

FIG. 8 is a flow diagram illustrating an example of a method for executing software.

FIG. 9 is a flow diagram illustrating an example of a method for executing software.

FIG. 10 is a flow diagram illustrating an example of a method for executing software.

FIG. 11 is a flow diagram illustrating an example of a method for executing software.

FIG. 12 is a diagram illustrating an example of a device illustrated in FIGS. 1 through 5.

FIG. 13 is a flow diagram illustrating an example of a method for causing software to be executed.

FIG. 14 is a flow diagram illustrating an example of a method for causing software to be executed.

FIG. 15 is a flow diagram illustrating an example of a method for causing software to be executed.